Polyphase FFT Channelizers: A Detailed Derivation

Philipp Stanner *

September 2023

Abstract

Slicing a signal's spectrum into channels is a common requirement in telecommunications and signal processing. Polyphase FFT Channelizers are digital structures that are very efficient at this task. Consequently, they are widely used in various areas of digital signal processing. Since their development in 1974, detailed derivations have been scarce and existing ones are often very compact or focus on optimized versions of the channelizer, which are harder to grasp. This paper details a new derivation for the non-optimized, basic channelizer as well as the optimized version with a commutator and one with a double commutator for reduced aliasing. The reader requires only basic system theory knowledge.

1 Introduction

Polyphase FFT Channelizers, first described in 1974 by Bellanger and Daguet [1], require far less calculation effort channelizing signals than a cascade of single filters could [2]. They can construct channelizers for television [3], receivers for FM [4] and OFDM signals [5] and have been suggested as key components for generic wideband receivers [6, 7]. Furthermore, they are also a valuable asset for wavelet analysis [2].

Unfortunately, complete and detailed derivations for these complicated structures have been scarce. For instance, research papers [8] and [6] discuss optimized versions of the channelizer, with the rate reduction being the first block in the flowgraph. Naturally, such optimized versions are harder to understand. Conversely, this paper presents both fundamentals and optimizations. Section 2 introduces the necessary mathematical foundations and system theory knowledge. Afterwards, Section 3 explains channelizers' fundamentals and describes how a reused (modulated) model filter can construct a filter bank; Section 4 provides a detailed explanation of how polyphase decomposition of said model filter constructs the Polyphase FFT Channelizer. Finally, Section 5 shows how the resulting non-optimized structure can be optimized by moving the sample rate reduction step to the start of the system's processing chain. Additionally, Section 5 establishes foundational knowledge necessary to implement further optimizations and introduces one such highly optimized version, a channelizer with a reduced sample rate reduction and, consequently, reduced self-aliasing.

*Correspondence: stanner@posteo.de

License: This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. For more information, see `https://creativecommons.org`

2 Mathematical Background

This section shall briefly establish the necessary mathematical and system theory knowledge background.

2.1 Conventions

All signals in this work are represented as complex numbers, while all filter taps are real-valued if not stated otherwise. Functions, variables and sequences are not marked explicitly to be complex.

Throughout this paper the following mathematical markers are used:

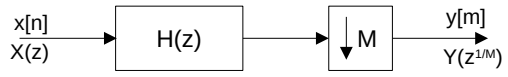
- Exponent (p), i.e., $F_\lambda^{(\mathrm{p})}$, identifies a function to be a polyphase component of the original function F.
- The caret symbol, i.e., $\hat{X}_\lambda$, denotes signals as filtered by a polyphase-filter, without themselves being polyphases.
- The tilde, i.e., $\tilde{y}$, marks temporary functions or variables that are created for formal reasons.

Frequencies, including the sampling frequency Ω_S and the Nyquist Frequency Ω_N, are presented as *normalized frequencies* Ω [9, 2, 10], allowing for the formulation of equations independent of sampling time T_S:

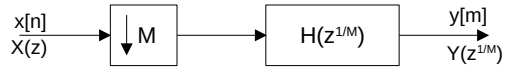
$$\Omega = \omega T_\mathrm{S} \tag{1}$$

$$\text{with } \Omega_\mathrm{S} = \omega_\mathrm{S} T_\mathrm{S} = 2\pi \text{ and } \Omega_\mathrm{N} = \pi \tag{2}$$

When sample rate *changes* are discussed, however, f_S is preferred because a normalized frequency would be less intuitive in this context.



(a) *Filtering with subsequent reduction.*



(b) *Reduction with subsequent filtering.*

Figure 1: *Equivalence of two positions of rate reductions. The signals X and Y are identical for both versions (a) and (b), but the filter changes.*

2.2 Sample Rate Changes

It is preferable to keep sample rates f_S as low as possible in digital systems because electronics used in modern digital circuits consume more energy with increasing clock frequency [9, 2]. Reducing f_S before a signal processing block such as a filter is especially useful as it reduces the number of operations per second, resulting in a far more efficient system structure. This is possible thanks to the *Noble Identity*, which defines several equivalences of sample rate change positions [2, 11], some of which are presented here.

A conventional approach to rate reduction is reducing the rate at the exit of a system, as shown in Fig. 1a. However, it is possible to reduce f_S before a signal processing block by using the equivalence from Fig. 1b. As shown, moving the rate reduction requires that the system's transfer function is either expanded or reduced by M, depending on which system is the reference system. The relationship between sample rate changes in time and frequency domain is given by [2, 11, 10]:

$$F(z) \bullet\!\!-\!\!\circ\, f[n] \tag{3}$$

$$F\left(z^{\frac{1}{M}}\right) \bullet\!\!-\!\!\circ\, \tilde{f}[m] = f[nM] \tag{4}$$

$$F\left(z^{M}\right) \bullet\!\!-\!\!\circ\, \tilde{f}[m] = \begin{cases} f\left[\frac{n}{M}\right] & \text{if } nM \text{ even} \\ 0 & \text{else} \end{cases} \tag{5}$$

Conversely, if the system in Fig. 1b was the reference with a function $H(z)$, the filter function in Fig. 1a has to be $H(z^M)$.

Equation (4) shows that downsampling a signal by a factor of M is achieved by simply keeping every Mth sample and dropping those in between; (5) shows that upsampling by a factor of M is achieved by inserting $M-1$ zeroes between the original samples.

2.3 Polyphase Representation of Systems and Signals

This section describes the polyphase representations of signals, summarizing [10] and [2].

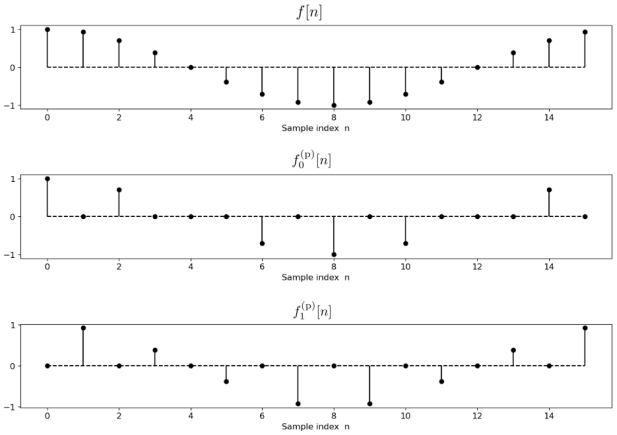


Figure 2: *A full period of a digitized cosinus $f[n]$ split into two polyphases $f_0^{(p)}[n]$ and $f_1^{(p)}[n]$. By adding up the polyphases the original signal is reconstructed.*

A polyphase representation is a parallel representation that can be applied to all digital signals. The conventional serial form of a system, with one input and one output, is referred to as serial representation in this document. A generic signal $f[n]$ can be split into an arbitrary number M of polyphases $f_\lambda^{(\mathrm{p})}$:

$$f[n] = \sum_{\lambda=0}^{M-1} f_\lambda^{(\mathrm{p})}[n] \tag{6}$$

The exponent (p) indicates that the associated term is a polyphase partition of the original signal. Any polyphase is characterized as follows, with M specifying the number of polyphases and $\lambda = 0, 1, 2, \ldots, M-1$ determining the time delay (phase):

$$f_\lambda^{(\mathrm{p})}[n] = \begin{cases} f[n+\lambda] & \text{if } (n+\lambda) \bmod M = 0 \\ 0 & \text{else} \end{cases} \tag{7}$$

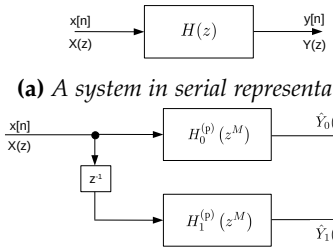
The corresponding descriptions in the frequency domain are:

$$F(z) = \sum_{\lambda=0}^{M-1} z^{-\lambda} F_\lambda^{(\mathrm{p})}(z^M) \tag{8}$$

$$F_\lambda^{(\mathrm{p})}(z^M) = \sum_{n=-\infty}^{\infty} f[nM+\lambda]\, z^{-nM} \tag{9}$$

Figure 2 shows a polyphase representation of the signal $f[n]$ for $M = 2$. Polyphase representation separates the original signal $f[n]$ into M polyphase-signals $f_\lambda^{(\mathrm{p})}$, each with a delay of 1 in respect to its predecessor. Each of the polyphases has to be sampled up by M, as z^M in (8) indicates.

The sampling process has a phase (meaning: delay) of λ equivalent to the polyphase's index ranging from 0 to $M-1$. The process of upsampling combined with the delay results in the zeroes and shift necessary for the polyphases to reconstruct the original signal again when summed up (see (6) and (8)).



(a) *A system in serial representation.*

(b) *The system in polyphase representation with $M = 2$.*

Figure 3: *An FIR-System in serial representation and polyphase representation. In polyphase representation, the filter taps are intertwined with zeroes, as indicated by z^M. Signals $x[n]$ and $y[n]$ are identical for both versions. The delay block z^{-1} implements the necessary phase of $\lambda = 1$.*

Likewise, every FIR-System can be split into a polyphase representation. This can be conceptualized by the fact that an FIR-System is, by definition, fully characterized by its impulse response, which is identical to the filter taps [9]. By splitting the impulse response into a polyphase representation, the system itself can be split. Consequently, a filter in serial representation as depicted in Fig. 3a can be divided into its polyphase representation and calculated in parallel, as clarified in Fig. 3b. As a result of this, the original filter's taps are distributed among the polyphase filters and are filled with intermediate zeroes, as described in (7) and illustrated by Fig. 2. The serial representation is re-entered by summing up the polyphases as described in (6) and (8). In an implementation, this is achieved through a summation point, as clarified in Fig. 3b.

The signals $\hat{Y}_\lambda(z)$ in Fig. 3b are polyphase-filtered subsignals:

$$\hat{Y}_\lambda(z) = X(z)\, H_\lambda^{(\mathrm{p})}\left(z^M\right) \tag{10}$$

As these signals are not upsampled, their function term contains z instead of z^M.

3 Filter Banks and Channelizers

A channelizer is a system that splits a spectrum into a variable number of smaller, consecutive subbands. These subbands are then shifted downwards (i.e., demodulated) to form baseband signals.

3.1 Fundamentals

At its core a channelizer can be imagined and implemented as a cascade of filters cutting out the desired spectrum for every output channel. A cascade of filters is referred to as a *filter bank*. Such a system is depicted in Fig. 4. A filter bank which slices a spectrum into smaller subbands is called an analytical filter bank. For simplicity's sake, filter banks are depicted with only four channels within this document. However, the following derivation does work accordingly for larger banks.

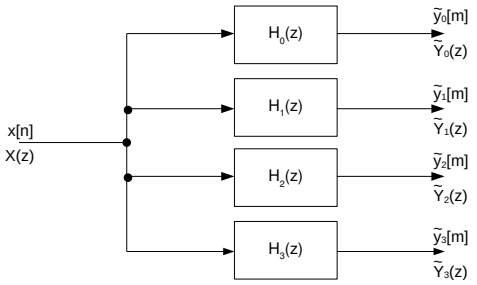


Figure 4: *An analytical filter bank consisting of four channels.*

Uniform Filter Banks A uniform filter bank is a filter bank with equidistant channels in which all channels have the same bandwidth (such as depicted in Fig. 5). The sum of all bandwidths amounts to the total available bandwidth $B = \Omega_\mathrm{S}$:

$$\sum_{k=0}^{M-1} B_k = B = \Omega_\mathrm{S} = 2\pi \tag{11}$$

$$B_k = \frac{B}{M} \tag{12}$$

The channels' center frequencies Ω_k are consequently placed in the middle of their bandwidth share:

$$\Omega_k = \frac{\Omega_\mathrm{S}}{M} k = \frac{2\pi}{M} k \tag{13}$$

To slice the input spectrum continuously, the channel borders, and, therefore, the filters' cutoff frequencies should ideally be positioned at

$$\Omega_\mathrm{k} \mp \frac{\Omega_\mathrm{N}}{M}. \tag{14}$$

The transfer functions shown in Fig. 5 adhere to the aforementioned rules.

Filter Banks as a Channelizer's Groundlayer Figure 6 shows a filter bank functioning as a channelizer's core component. As seen in Fig. 5, the Filters $H_\mathrm{k}(z)$ have to be bandpass filters with different center frequencies Ω_k, retrieving the desired subspectra of the input signal $X(z)$.

Similar to what classical amplitude demodulation accomplishes, the demodulation blocks D_k are responsible for shifting the filtered bandpass signals $\tilde{Y}_\mathrm{k}$ downwards by their center frequency Ω_k, turning them into the baseband signals Y_k. In the time domain this demodulation is implemented in form of a multiplication with a rotating pointer [9]:

$$D_k(z) \bullet\!\!-\!\!\circ\; d_k[n] = e^{-j\frac{2\pi}{M}kn} \tag{15}$$

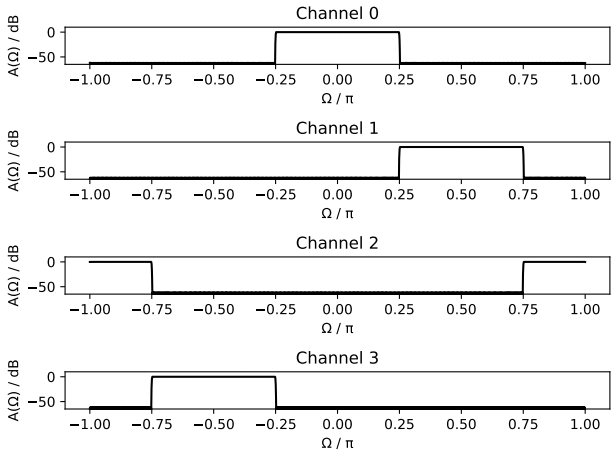


Figure 5: *Example of a 4-channel uniform filter bank transfer function, implemented by the filter bank of Fig. 4. Each filter cuts out a fourth of the available complex bandwidth.*

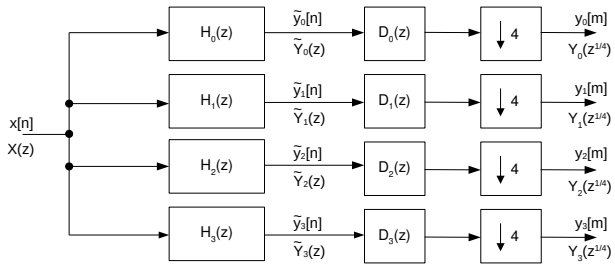


Figure 6: *The filter bank previously illustrated in Fig. 4, here extended to be a channelizer. The demodulation blocks $D_k(z)$ turn the bandpass signals $\tilde{Y}_k(z)$ into baseband signals, which then are permissibly sampled down by a factor of four.*

3.2 Downsampling

The output signals in Fig. 6 each have the bandwidth B/M, as defined in (12). As these signals are now baseband signals, their maximum and minimum frequencies are

$$\pm\Omega_{\max} = \pm\frac{\Omega_N}{M}. \tag{16}$$

Consequently, rate reduction by a factor M is possible without violating the Nyquist Theorem:

$$f_S' = \frac{f_S}{M} \tag{17}$$

While sample rate reductions are widely used in practical implementations they are not necessary to understand the non-optimized, basic version of a Polyphase FFT Channelizer. Therefore, further details on rate reductions are provided at a later point in the document regarding optimizations in Section 5.

3.3 Disadvantages of Manual Filter Bank Design

Designing and operating the types of filter banks that have been presented so far is difficult because of the following reasons:

- M filters have to be designed manually.
- M outputs must be calculated during runtime, resulting in the input sequence $x[n]$ having to be convolved with all M filter tap sequences $h_k[n]$.

This paper only considers uniform filter banks. As uniform filter banks consist of equally sized channels only varying in their center frequencies, it is reasonable to dedicate efforts towards reusing a single channel filter.

3.4 Modulated Filter Banks

We utilize the properties of the Z-Transform to derive how shifting a filter's transfer function $H(z)$ corresponds to changes on the filter taps $h[n]$: $H(z) \bullet\!\!-\!\!\circ h[n]$. By shifting a single Model Filter, all other channel filters are obtainable [2], which enables to design a uniform filter bank. This model filter is a lowpass filter that is transformed into bandpass filters by shifting it in the frequency domain.

Modulating Filter Coefficients The *modulation property*[1] of the Z-Transform is defined as follows [12, 9]:

$$F\left(\frac{z}{a}\right) \bullet\!\!-\!\!\circ\ a^n f[n] \quad \forall\, a \in \mathbb{C}\setminus\{0\} \tag{18}$$

[1] Sometimes also referred to as »scaling property«

If a represents a complex number $a = e^{-j\Omega}$, with Ω specifying an arbitrary angle (frequency), the exponential laws of mathematics show that a becomes a complex rotating pointer in the time domain:

$$a^n = \left(e^{-j\Omega}\right)^n = e^{-j\Omega n} \tag{19}$$

One could interpret this utilization of the property such that the filter coefficients start »rotating« in the complex plane at the frequency Ω. By scaling with (19), the filter's transfer function is shifted upwards by Ω:

$$H\left(\frac{z}{e^{-j\Omega}}\right) \bullet\!\!-\!\!\circ e^{-j\Omega n}\, h[n] = [h[0]e^{-j0\Omega},\ h[1]e^{-j1\Omega},\ h[2]e^{-j2\Omega}, \dots] \tag{20}$$

Example: Scaling the transfer function $H_0(z)$ of a lowpass filter serving as a model filter with $a = e^{-j\pi/2}$ and operating it at $f_S = 1000\,\text{Hz}$ transforms the lowpass into a bandpass with its center frequency at 250 Hz.

Uniform Filter Bank Design with Modulated Filters The center frequencies of all channels Ω_k are now placed as described by (13). Through (18), a single model filter H_0 is transformed into a cascade of bandpass filters with complex coefficients. In order to design a filter bank with the frequency response shown in Fig. 5, the taps for each of the four filters must be calculated using those of the model filter as a basis. The following equation describes how the filter taps for a channel k are computed:

$$h_k[n] = \left(e^{-j\frac{2\pi}{M}k}\right)^n h_0[n] \tag{21}$$

It is, therefore, possible to create a uniform filter bank by modulating a single model filter.

In the frequency domain representation of (21), we define the *modulated time discrete complex frequency* $\tilde{z}$ in the following form:

$$H_k(z) = H_0\left(z\, e^{j2\pi\frac{k}{M}}\right) =: H_0(\tilde{z}) \tag{22}$$

Designing a uniform filter bank through modulated filter coefficients has disadvantages, some of which already occurred in the conventional filter bank:

- It is de facto still necessary to design M filters: Each single channel filter's tap sequence has to be computed through (21).
- It is still required to compute M convolutions during runtime.
- The originally real-valued filter taps turn into complex-valued filter taps, resulting in more complicated convolution calculations.

Nevertheless, modulating a model filter is the first step on the way to a more efficient implementation.

4 Polyphase FFT Channelizers

Combining the knowledge from previous sections permits designing a highly efficient form of a channelizer, referred to as *Polyphase FFT Channelizer* or, briefly, *Polyphase Channelizer*.

4.1 Modulation-Independent Filter Coefficients

To derive the new structure, the single channel filter $H_0(z)$ of a channelizer as shown in Fig. 6 is chosen with the intention to use it as a universal filter by replacing the transfer function by the one described in (22). Hence, we utilize a model filter to derive all channel filters from it. This filter can now be interpreted as a filter responsible for cutting out an arbitrary channel, depending on the value of k. With adjustments to the transfer function, Fig. 3a could represent such a filter.

The universal filter $H_0(\tilde{z})$ is now **polyphase-decomposed into M polyphases** (compare with Fig. 3b) according to the rules defined in Section 2.3 – see especially (8) and (9). As defined previously, polyphase decomposition substitutes $\tilde{z}$ with $\tilde{z}^M$. Therefore, in the *modulated universal* channel filter, $\tilde{z}$ is expressed as $\left(z\exp(j2\pi\frac{k}{M})\right)^M$, and according to (8) the universal transfer function in polyphase representation for any channel k is

$$H_k(z) = \sum_{\lambda=0}^{M-1} \tilde{z}^{-\lambda} H_{0,\lambda}^{(p)}\left(\tilde{z}^M\right). \tag{23}$$

The extended form of the summands in (23) is

$$\tilde{z}^{-\lambda} H_{0,\lambda}^{(p)}\left(\tilde{z}^M\right) = \left(z\, e^{-j2\pi\frac{k}{M}}\right)^{-\lambda} H_{0,\lambda}^{(p)}\left(\left(z\, e^{j2\pi\frac{k}{M}}\right)^M\right). \tag{24}$$

The function argument of $H_{0,\lambda}^{(p)}$ in (24) evaluates to z^M for every channel k:

$$\left(z\, e^{j2\pi\frac{k}{M}}\right)^M = z^M e^{j2\pi k} = z^M \quad \text{for } k \in \mathbb{N}_0 \tag{25}$$

Thus, (24) can be simplified to

$$\begin{aligned}\tilde{z}^{-\lambda} H_{0,\lambda}^{(p)}\left(\tilde{z}^M\right) &= \left(z\, e^{-j2\pi\frac{k}{M}}\right)^{-\lambda} H_{0,\lambda}^{(p)}\left(z^M\right) \\ &= z^{-\lambda}\, H_{0,\lambda}^{(p)}\left(z^M\right)\, e^{j\frac{2\pi}{M}\lambda k}.\end{aligned} \tag{26}$$

Equation (26) shows that in polyphase representation the coefficients in the modulated filter now are **independent** *from any frequency shift and channel index k*. The coefficients do not rotate anymore as they previously did in (20).

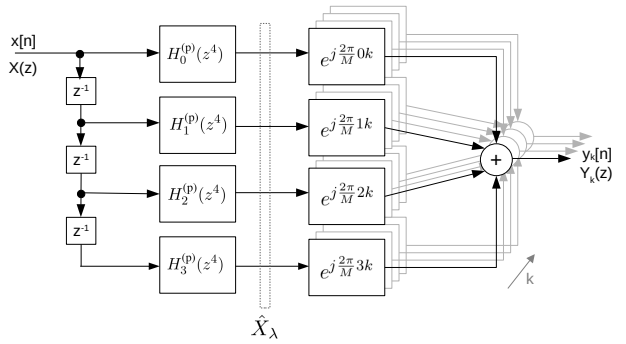


Figure 7: *A modulated universal channel filter in its polyphase representation with $M = 4$. The summation point serves entering serial representation again. M blocks with $\hat{X}_\lambda$ as inputs (highlighted in gray) are necessary to obtain the signals for all channels. Except for the value of k, each of the gray substructures is identical.*

Consequently, the simplified universal transfer function for an individual channel k is:

$$H_k(z) = \sum_{\lambda=0}^{M-1} z^{-\lambda} H_{0,\lambda}^{(\mathrm{p})}\left(z^M\right) e^{j\frac{2\pi}{M}\lambda k} \tag{27}$$

Equations are made more compact to (ultimately) obtain a term for the output signal. To do so, the intermediate polyphase-filtered signals $\hat{X}_\lambda$ (see Fig. 7) are defined to describe the part of the processed signal X which is common across *all* channels:

$$\hat{X}_\lambda(z) := X(z)\, z^{-\lambda}\, H_{0,\lambda}^{(\mathrm{p})}\left(z^M\right) \tag{28}$$

To obtain the output of a channel k, these intermediate signals have to be multiplied with the complex value $e^{j\frac{2\pi}{M}\lambda k}$. Afterwards, the paths are summed up to leave the filter's polyphase representation and reenter serial representation, just like in Fig. 3b. The output of an individual channel based on (28) according to (8) then results to:

$$Y_k(z) = \sum_{\lambda=0}^{M-1} \hat{X}_\lambda(z)\; e^{j\frac{2\pi}{M}\lambda k} \tag{29}$$

Equations (28) and (29) describe the flowgraph of the modulated universal channel filter in Fig. 7. This flowgraph illustrates a universal channel filter with coefficients that are now independent of k. Channel dependency occurs after the polyphase filters when the signals $\hat{X}_\lambda$ are multiplied with the channel-dependent complex pointers.

To implement the channelizer depicted in Fig. 6 using modulated filters, M of the complex pointer blocks on the right hand side of $\hat{X}_\lambda$ are necessary. These blocks are highlighted in gray in Fig. 7.

Implementing said pointer blocks manually would entail further implementation efforts. Fortunately, a ready-to-use algorithm implementing the desired operations is known since many decades.

4.2 FFT-Modulated Filter Banks

As was defined in (29), to retrieve the output signal $Y_\mathrm{k}(z)$ for a channel k, the polyphase-filtered intermediate signals $\hat{X}_\lambda$ have to be multiplied by the complex number $e^{j\frac{2\pi}{M}\lambda k}$ and then be added. Transforming (29) into the time domain results in the following representation:

$$y_k[n] = \sum_{\lambda=0}^{M-1} \hat{x}_\lambda[n]\, e^{j\frac{2\pi}{M}\lambda k} \quad \text{with } n = 0, 1, \ldots, \infty \tag{30}$$

Except for the missing factor $1/\mathrm{N}$, (30) is identical to the computation rule for the Inverse Discrete Fourier Transformation (IDFT)[2] [9]:

$$x_i = \frac{1}{N} \sum_{l=0}^{N-1} X_l\, e^{j\frac{2\pi}{N}li} \tag{31}$$

In (31), X_l denotes a frequency bin with index l, while x_i is the corresponding time value after transforming a discrete spectrum consisting of N bins back into a time-domain signal, resulting in N samples. The IDFT implements (30):

- X can be interpreted as $\hat{X}_\lambda$ from Figures 7 and 8.
- The sum of all polyphases λ in (30) can be interpreted as the sum over all rotation steps l in (31).
- In an instant of time n_0, a channel output sample y_k is equal to the output sample x_i of the IDFT.

Consequently:

$$\begin{aligned} l \equiv \lambda,\; i \equiv k \\ y_k = x_i \\ M = N \end{aligned} \tag{32}$$

The Polyphase FFT Channelizer Considering all the previously introduced aspects, the new structure in Fig. 8 illustrates an implementation where the IDFT block replaces the previously necessary multiplications with the complex pointers, as well as the summation point to exit polyphase representation, both depicted in Fig. 7. The flowgraph includes blocks for demodulation and sample rate reduction, as discussed in Section 3.1.

It is important to note that there are differences to the earlier presented version of the channelizer depicted in Fig. 6. The four filters here are not the channel filters. They are the *polyphase split of a single model filter*. As a result of this, the filter functions H are indexed with the polyphase index λ instead of the channel index k from here on. The necessity of

[2] The typically used indices n, k have been replaced with i, l to prevent confusion with the time index n and channel index k.

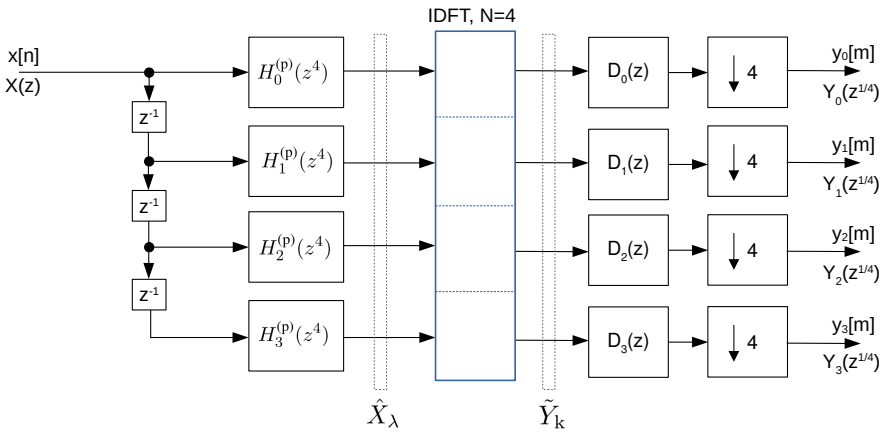


Figure 8: *The derived channelizer. The four filters are polyphase partitions of the model filter. The signals $\tilde{Y}_k$ are the bandpass filtered channels, before they are shifted downwards into baseband.*

this index-change in general is also clarified in Fig. 8 ($\hat{X}_\lambda$ and $\tilde{Y}_\mathrm{k}$).

After the IDFT, the separate channels can be demodulated and, as their maximum frequency is now limited, the sample rate can be reduced. After rate reduction, the outputs' Nyquist Frequencies are all $f'_\mathrm{N} = \frac{1}{M} f_\mathrm{N}$.

Streamed IDFT Strictly speaking, the filter bank as described by (30) operates on mathematical sequences, whereas (Inverse) Discrete Fourier Transformations operate on data vectors. To obtain the output vector $\hat{Y}_{n_0} = [\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_{M-1}]$ for a point in time n_0, it is necessary to compute (30) for each k. Therefore, this procedure needs to be repeated in intervals of T_S to obtain an output stream. One could interpret the procedure as a »streamed IDFT« where the input vector changes with every increment of n:

$$x_i[n] = \frac{1}{N} \sum_{l=0}^{N-1} X_l[n]\, e^{j\frac{2\pi}{N} li} \tag{33}$$

The streaming aspect is visualized in Fig. 8. The polyphase-branches supply the IDFT with M samples. The IDFT's payload is updated and calculations are performed with every increment of n.

4.3 Derivation Summary

The following list summarizes the significant steps in deriving the polyphase channelizer:

1. A model filter is modulated with channel index k acting as a variable for specifying the target channel frequency. (22).
2. The modulated model filter is split into its polyphase representation. (23), (24).
3. This split results in equal filter coefficients across all channels k. (25), (26).
4. By reshaping the modulated polyphase delays $\tilde{z}^{-\lambda}$ from (23) a complex pointer is produced that depends on both variables k and λ. (26).
5. This pointer combined with the sum (Fig. 7) necessary to leave polyphase representation again results in equivalence (except for 1/N) with the IDFT-Algorithm. (32).

It is important to highlight that the IDFT does *not* modulate the filter coefficients – it implements the modulation of the polyphase delays $\tilde{z}^{-\lambda}$ in (23). As previously shown through (25), modulating the coefficients is no longer necessary, as they became independent from any channel index. All of the steps listed above, in particular entering and leaving polyphase representation, bring forth a mathematical expression equivalent to the IDFT.

4.4 Advantages

The presented structure provides a variety of advantages. For instance, the signals $\hat{X}_\lambda$ are identical across all channels and, therefore, only have to be calculated once. Furthermore, the IDFT is among the most essential mathematical operations in communication technology. Due to their long history of usage, optimized DFT-libraries exist for virtually all relevant digital systems. These highly optimized implementations make use of an algorithm referred to as *Fast Fourier Transformation (FFT)*. Available FFT-Libraries are likely to be far more efficient than custom implementations of the sum described in (30) would be. This is why polyphase channelizers are predominantly implemented with an (I)FFT. Consequently, they are often also referred to as *Polyphase FFT Channelizers*.

Additionally, design effort is reduced as the engineer has to design only one low pass filter (the model filter) and specify the number of channels. At this point, however, the total number of filter taps

and, consequently, filter calculation effort is not reduced: The model filter's taps are distributed to the polyphases, but have to be filled with zeroes (see Fig. 2 and (5)), resulting in the same total number of taps. Section 5 shows how these zero-taps can be eliminated.

4.5 IDFT or DFT

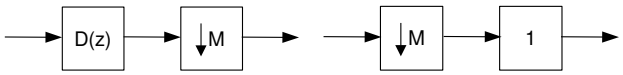
Relevant sources (e.g., Fliege [2] and Harris [6, 8]) always present the channelizers with the IDFT. However, DFT and IDFT are entirely interchangeable. The key to understanding this lies in Section 3.4: It was implicitly decided to shift the model filter upwards by setting variable a in (18) to the negative pointer $e^{-j\Omega_k}$. If the model filter were to be shifted *downwards* with a positive pointer $e^{j\Omega_k}$, and the rest of the derivation is carried out in the same manner as described in previous sections, the result in Fig. 8 would contain the DFT instead of the IDFT. Due to the different shifting direction, this would cause the spectral shares of channels one and three in Fig. 5 to switch positions.

The DFT would implement the desired modulation algorithm exactly, because it lacks the damping by a factor $1/N$ which occurs in the IDFT [9]. As this difference is not very significant, this paper uses the IDFT to maintain congruency with previous publications.

Engineers presumably often prefer the IDFT because it suits the following perception better: a channelizer slices a bandwidth into channels and demodulates them, so they become baseband signals. Hence, one might interpret the IDFT's input data as signals separated in the frequency domain and the output as time domain data. Thus, the IDFT acts as a separation mechanism between frequency domain on the left and the time domain on the right [1] (compare with Fig. 12). Using a DFT instead would not allow for this interpretation to be valid, although the system's behavior would be almost identical. However, this interpretation is purely aesthetical and neither necessary to design nor to understand the channelizer. Section 5.2 provides further details on the IDFT's interpretation.

5 Polyphase Channelizer Optimizations

This last section introduces the most basic form of an optimized channelizer, the »Commutated Polyphase Channelizer« and a special version with a double commutator that has reduced aliasing compared to the primer.



(a) *Demodulation followed by rate reduction.* **(b)** *Rate reduction moved to the D-block's left.*

Figure 9: *The rate reduction rule from Section 2.2 applied to the demodulation blocks. The D-Block's transfer function becomes equal to 1 and, therefore, obsolete.*

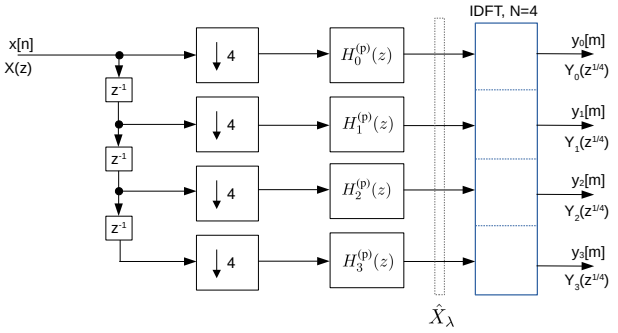


Figure 10: *An intermediate step to the optimized version: the rate reduction blocks have first been moved over the D-Blocks, making the later obsolete. In a second step, they have been moved over the polyphase filters, causing $H_\lambda^{(p)}(z^4)$ to change to $H_\lambda^{(p)}(z)$.*

5.1 Commutated Polyphase Channelizers

As mentioned in Chap. 1, practical implementations typically move the rate reduction to the system's start to reduce calculation effort. Specialized channelizers implementing rate reductions of factors M, $\frac{1}{2}M$ and $\frac{3}{4}M$ have been described [13]. As a first introduction to the general concept, this section describes how the presented channelizer with a rate reduction of factor M can be optimized. This section starts off by providing an introduction to the general concept of commutated polyphase channelizers and describes the necessary optimization steps for the presented channelizer with a rate reduction of factor M. A justification is given for why rate reductions unequal to M are advantageous in the first place – afterwards, one such implementation, with a rate reduction of a factor $M/2$, is introduced.

The following list enumerates the steps necessary to optimize the channelizer from Fig. 8.

1. When the D-Blocks are implemented as rotating pointers as in (15), shifting the rate reduction to their left results in the pointers becoming equal to 1, as only every Mth sample is kept [14]:

$$y_k[m] = \tilde{y}_k[n]\, e^{j\frac{2\pi}{M}kn} \overset{\downarrow M}{\Longrightarrow} y_k[m] = \tilde{y}_k[Mn] e^{j2\pi\, kn} = \tilde{y}_k[Mn] \quad (34)$$

   The disappearance of the D-Blocks is visualized by Fig. 9.

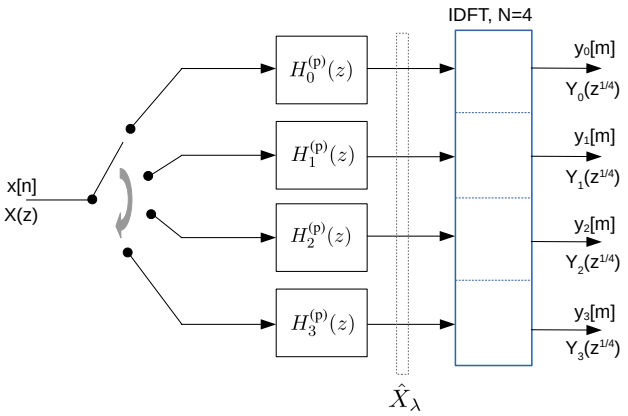


Figure 11: *An optimized version of the channelizer. The commutator moves downwards with a frequency of f_S, starting at the top again after reaching the last position.*

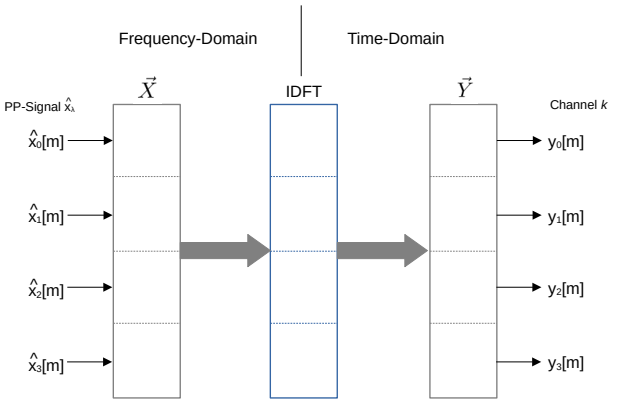


Figure 12: *The IDFT can be interpreted to act as a mechanism splitting the system into a frequency and a time domain. In their respective domains, input and output can be conceptualized as vectors. Thus, $\vec{X}$ and $\vec{Y}$ are corresponding with each other.*

2. As the IDFT is only a sum and multiplication with (per channel k) constant complex numbers, the rate reduction can be moved over it without further efforts.
3. When the rate reduction is further moved to the polyphase-filters' left, those filters' expansion by factor M disappears, as detailed in Section 2.2:
$$H_\lambda^{(p)}(z^M) \overset{\downarrow M}{\Longrightarrow} H_\lambda^{(p)}(z) \tag{35}$$
This simplifies computing the filter convolution, as coefficients equal to zero disappear and the number of taps is reduced by a factor of M (see (5) and (4)).
4. Now, the rate reductions are directly at the system's start, on the right side of the z^{-1} delays, as portrayed in Fig. 10. The rate reductions can now be combined with the delays to form a commutator.

Ultimately, the steps above result in the *Commutated Polyphase Channelizer* depicted in Fig. 11. The great advantage of this structure compared to its predecessors is a substantial reduction in calculation costs: the number of taps is reduced by M.

Performance Fliege [2] calculates that a channelizer such as the one illustrated in Fig. 11, with 32 channels and a model filter with 256 taps reduces calculation effort by a factor 1046 compared to a structure with 32 independent bandpass filters with the same number of taps. In general, the efficiency can be attributed to the following points:

- Instead of M, only one filter has to be calculated during runtime. This filter is split into its polyphases.
- The polyphase split allows parallel filter calculation, which utilizes resources such as processor cores more efficiently.
- Moving the rate reduction to the system's start causes
  - the system to be operated at a lower frequency.
  - the expansion z^M to collapse to z. Hence, there are no more filter taps with zero value anymore, which, unfortunately, had to be considered when calculating the filter convolution.
- The IDFT can be implemented as an IFFT. FFT-implementations are available virtually everywhere and can be trusted to be very performant.

There seems to be a common consensus that the most efficient mechanism to split a signal into an even number M of equally large frequency slices is a Commutated Polyphase FFT Channelizer. [2, 15, 8, 16]

5.2 IDFT as a Gateway to DFT-Properties

For example when using reduction rates other than M, optimized versions may utilize another mathematical law to move rate reduction-related operations to the left side of the IDFT. As visualized in Fig. 12, the IDFT permits viewing the channelizer as a system separated into frequency domain (on the left) and time domain (on the right). Thus, one can interpret the transform's input bins as an input vector $\vec{X}$ and the output samples as an output vector $\vec{Y}$. As the figure establishes, through the IDFT, the two vectors are correspondences to each other:

$$\vec{X} \bullet\!\!-\!\!\circ\, \vec{Y} \tag{36}$$

Therefore, all changes to $\vec{Y}$ can be replaced by changes to $\vec{X}$ and vice versa, if there is an associated property for DFTs. The following section shows how the frequency shift property helps transforming rate

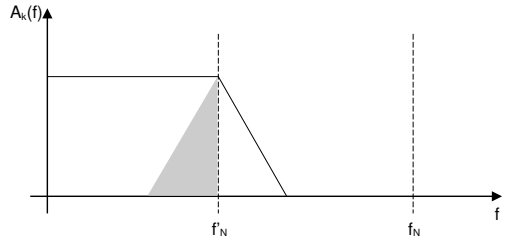
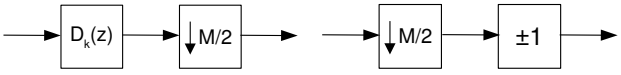


Figure 13: *Symbolic depiction of how a channel's transfer function distorts itself after reducing the sample rate. The gray area is the mirrored part of the transfer function exceeding the new Nyquist Frequency* $f'_N = \frac{1}{M} f_N$.



(a) *Demodulation followed by rate reduction.*

(b) *Rate reduction moved to the D-block's left.*

Figure 14: *Moving the rate reduction of* $M/2$ *as previously in Fig. 9 causes the reduction to become a multiplication by* ± 1.

reduction-related multiplications on $\vec{Y}$ into bin-swap operations on $\vec{X}$. This way, an efficient channelizer that implements a rate reduction of a factor $M/2$ gets constructed [8].

5.3 Twice Commutated Channelizer

The so far presented channelizer has one major disadvantage, which is depicted in Fig. 13: It is critically sampled. As filters with infinitely steep flanks are impossible to implement, a part of the filter's transfer function exceeds the new Nyquist Frequency f'_N. Thus, channels in a channelizer with a rate reduction of a factor of M distort themselves. Effectively, this leads to aliasing frequencies, being stronger near f'_N.

Depending on the available flank steepness, aliasing might or might not be a problem. In any case, if increasing the flank steepness is not an option, the problem can be lessened through a rate reduction by a factor smaller than M. But, as a strong rate reduction is desirable due to reduced energy consumption and computation cost, it is also not a desirable way not to decrease the rate at all. Therefore, a compromise is adopted: The rate is reduced, but instead of choosing the maximum reduction by factor M, the second strongest possible reduction by factor $M/2$ is applied. The following sections show how such a reduction can be implemented efficiently.

Demodulation Once again, the goal is to eliminate the demodulation blocks from Fig. 8 by moving the rate reduction over them. Moving the reduction by $\tilde{M} = M/2$ over the blocks delivers the following computation rule to get demodulated output samples:

$$y_k[n] = \hat{y}_k[n]\, e^{j\frac{2\pi}{M}kn} \implies y_k[n] = \hat{y}_k[\tilde{M}n]\, e^{j\pi kn} \quad (37)$$

$$\text{with} \quad e^{j\pi kn} = \begin{cases} +1 & \text{if } kn \text{ even} \\ -1 & \text{else} \end{cases} \quad (38)$$

Note that the sign of $\hat{y}$ depends on both k and n. Fig. 14 shows how shifting the rate reduction over the D-Blocks works in principle. Figure 15 shows how shifting rate reductions by a factor $M/2$ to the left in the channelizer in Fig. 8 changes the flowgraph.

Figure 16 and Table 1 clarify Equations (37) and (38). As illustrated, the demodulation can be implemented as a multiplication with -1 for all uneven channel indices k and uneven running indices n.

For the next step, it is helpful to view the IDFT as a separator between frequency and time domain with input and output vectors, as introduced in Section 5.2. While replacing the D-Blocks with comparatively simple multiplications with ± 1 is already an improvement, there is room for further optimization. The separation in frequency and time domain allows utilization of the *frequency shift property* for

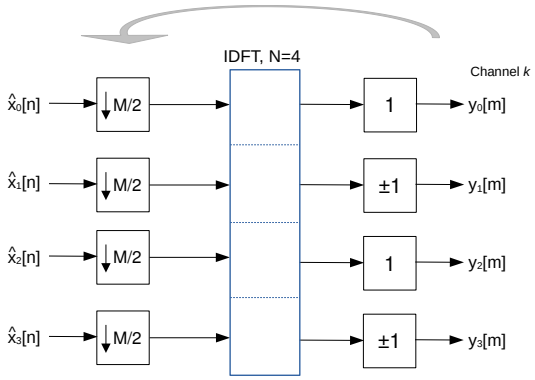


Figure 15: *Moving the rate reduction to the left turns the D-Blocks into multiplications with* ± 1.

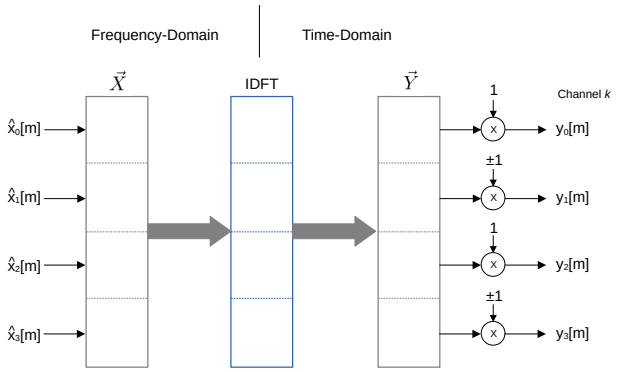


Figure 16: *Vector-based representation of Fig. 15 (compare also with Fig. 12).* $\vec{X}$ *(following the rate reduction) contains the payload getting transformed by the IDFT,* $\vec{Y}$ *is the output vector. To implement the demodulation, elements in* $\vec{Y}$ *with uneven channel index k must be multiplied with* ± 1. *The structure's inputs* $\hat{x}_\lambda$ *are the polyphase-filtered polyphase signals (compare Fig. 8).*

	n				
k	0	1	2	3	...
0	1	1	1	1	...
1	1	-1	1	-1	...
2	1	1	1	1	...
3	1	-1	1	-1	...

Table 1: *Time visualization of Equation* (38). *Samples of uneven channels k are multiplied with* -1 *for uneven values of n. Depicted are the states that the multipliers in Fig. 16 take over time.*

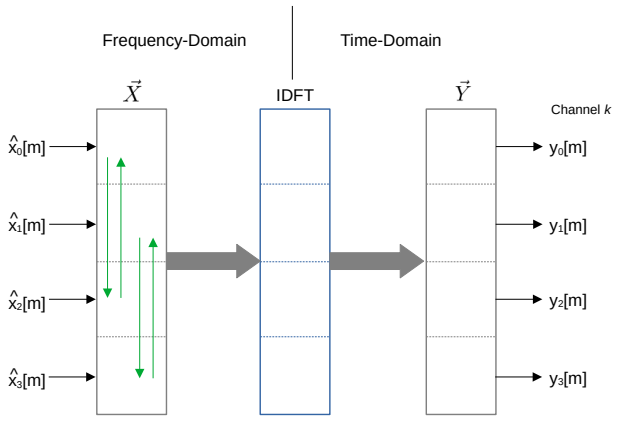


Figure 17: *Moving the multiplications in the time domain with* ± 1 *also to the IFFT's left is equivalent to switching* $\vec{X}$*'s samples (green arrows) in the frequency domain. This structure's inputs and outputs are identical to the ones in Fig. 16. The multiplication with* ± 1 *is equivalent to swapping the upper and lower half of* $\vec{X}$.

time-discrete Fourier Transformations [9]

$$F(\omega e^{j\Omega}) \bullet\!\!-\!\!\circ\, e^{j\Omega} f[n]. \tag{39}$$

Looking at Figures 16 and 17, one can interpret vectors $\vec{X}$ and $\vec{Y}$ to be corresponding with each other through this property. Hereby, $\vec{X}$ is seen as a vector containing frequency bins, and $\vec{Y}$ as one containing samples in time domain. By setting Ω to 2π or π, respectively, the complex numbers in the property are set to implement the desired multiplication with ± 1:

$$1 = e^{j2\pi} = e^{j\Omega_S}$$
$$-1 = e^{j\pi} = e^{j\frac{1}{2}\Omega_S}$$

The property permits shifting the multiplication from the right side of the IDFT to the left side. It shows that multiplying every second sample in the output vector with -1 (in the time domain) is equal to applying the maximum possible frequency change in the input vector. Hence, the frequency bins have to be shifted in their vector as far as possible.

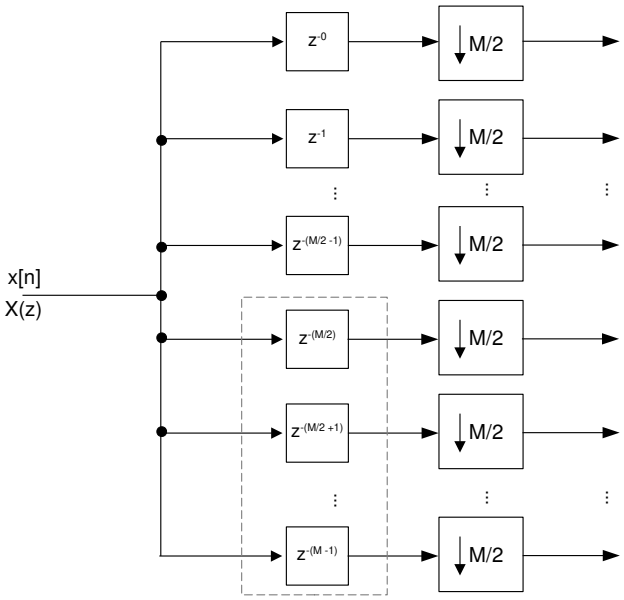


Figure 18: *The rate reductions on the right hand side of the polyphase delays. Here presented in absolute numbers and for a general bank of width M. The marked delays will be split in the next step.*

In summary, the demodulation of a filter bank with the sample rate reduced by M½ can be implemented by shifting the bins of the input vector in front of the IDFT. As Fig. 17 illustrates, this results in swapping the upper and lower half of the buffer. [8]

Furthermore, Table 1 establishes that the swap operation in Fig. 17 is only performed for uneven values of n.

Both implementations are in theory equivalent. However, it is reasonable to predict higher efficiency with the presented method when implemented, i.e., in FPGAs: Already available data need only be redirected into a different memory cell, instead of requiring a costly multiplication. As digital numbers are often implemented using *Two's Complement Encoding*, it is typically not possible to implement the multiplication by ± 1 by merely toggling an algebraic sign bit [17]. Therefore, the presented optimization is desirable.

Double Commutator In this next step, the sample rate reduction is moved to the left-most position, similarly as it was done previously in Equation (34) and Fig. 10. Moving the reduction by a factor of M½ over the polyphase-filters causes the expansion to collapse to

$$H_\lambda^{(p)}(z^M) \implies H_\lambda^{(p)}(z^2).$$

Finally, the rate reduction is at the left most position, on the right-hand side of the polyphase delays. For the last optimization step, a different representation

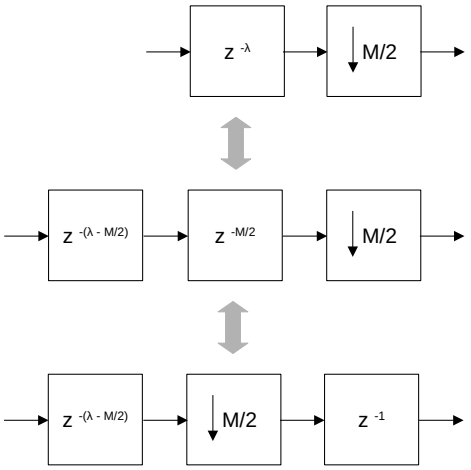


Figure 19: *Using equivalences to move the rate reduction over a part of the delay.*

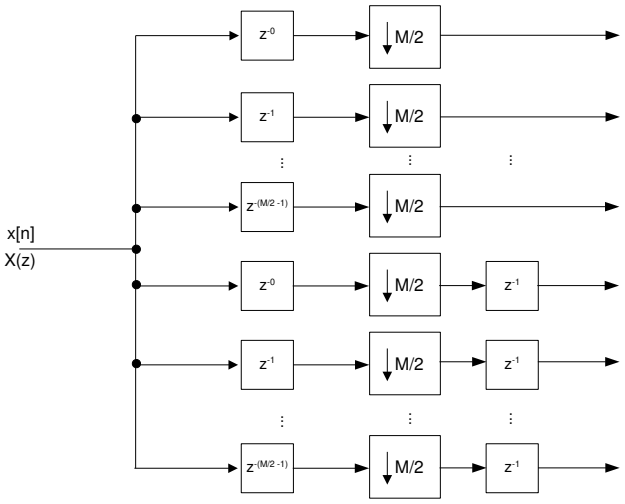


Figure 20: *Moving the rate reduction by $\frac{M}{2}$ over the $z^{-\frac{M}{2}}$ delays causes the latter to turn into z^{-1} delays. Now, upper and lower halves of the flow graph are identical on the rate reduction's left.*

is selected: Figure 18 presents this intermediate state of the structure for a bank of width M. The delays are presented in absolute numbers instead of building them with consecutive z^{-1} delays. The entire structure is conceptually split into two halves, with their border between $z^{-\frac{M}{2}-1}$ and $z^{-\frac{M}{2}}$. The lower half of the delays then gets split by extracting $z^{-\frac{M}{2}}$:

$$z^{-\lambda} = z^{-(\lambda-\frac{M}{2}+\frac{M}{2})} = z^{-(\lambda-\frac{M}{2})}\, z^{-\frac{M}{2}} \qquad (40)$$

This delay extraction is visualized in Fig. 19. Once more, we apply the rules from Section 2.2: the rate reductions in the lower half of Fig. 18 are moved to the left. This causes the delays to become equally delaying by a factor of 1. Figure 20 shows the intermediate result of this step. With this figure, the derivation is at a very similar point as it was in Fig. 10: it is now possible to replace the delays and rate reductions with a commutator. In this case, however, two commutators are necessary. The upper and lower halves of the structure differ in the lower half's values being delayed by one time step.

The final result can be seen in Fig. 21, which shows the derived structure for $M = 4$. As mentioned, the buffer switching portrayed in Fig. 17 only occurs for uneven n, as Equation (38) demands. Figure 22 depicts the system's output spectra[3] after the system has been stirred up with a Dirac-Impulse. Now, the channel borders do not lie at $\pm\Omega_\mathrm{N}$ anymore, but at $\pm\frac{\Omega_\mathrm{N}}{2}$. Therefore, the self distortion portrayed in Fig. 13 has been lessened, while at the same time the sample rate is reduced before the filters' costly computations. The filters' expansion has also been reduced to z^2. The channelizer presented in Fig. 21 is the final result of the optimizations performed in this document.

6 Conclusion

The presented paper revisited Polyphase FFT Channelizers roughly fifty years after their first publication. It is the author's wish that this document supports other engineers in understanding these arguably very complicated structures.

Acknowledgment

Many thanks to Dr.-Ing. Johannes Obermaier, Prof. Dr.-Ing. Arne G. Striegler and Mr. Timothy Tuchman for their advice. Additional gratitude belongs to »Vel«, who created this LaTeX-Template.

References

[1] M.G. Bellanger and J.L. Daguet. "TDM-FDM Transmultiplexer: Digital Polyphase and FFT". In: *IEEE Trans. Commun., vol. 22, pp. 1199-1204* (1974).

[2] Norbert Fliege. *Multiratensignalverarbeitung*. B.G. Teubner Stuttgart, 1993.

[3] Fred Harris, X. Chen, and V. Elettra. "Wideband 160-channel polyphase filter bank cable TV channeliser". In: *IEEE* (2010).

[4] Fred Harris and Benjamin Egg. "Forming Narrowband Filters at a Fixed Sample Rate with Polyphase Down and Up Sampling Filters". In: *IEEE* (2007).

[3] With a model filter with an arbitrarily chosen damping factor

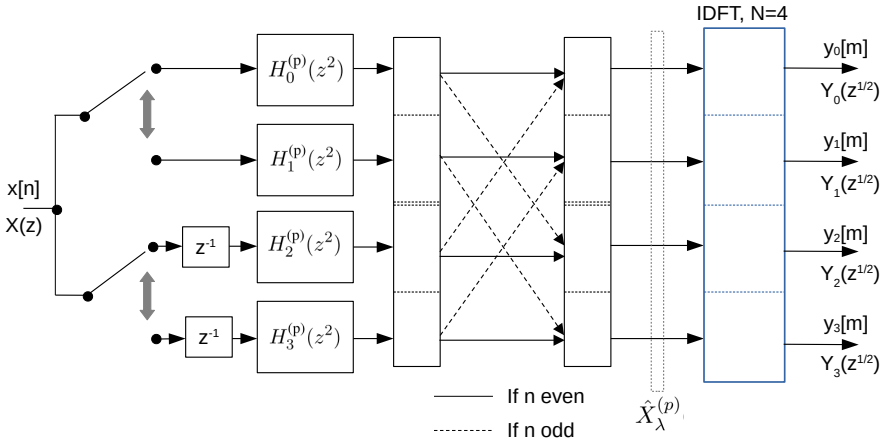


Figure 21: *The channelizer in Fig. 11 optimized to have reduced aliasing by reducing the sample rate only by factor M/2. Effectively, the upper and lower half of the buffers are swapped for uneven n.*

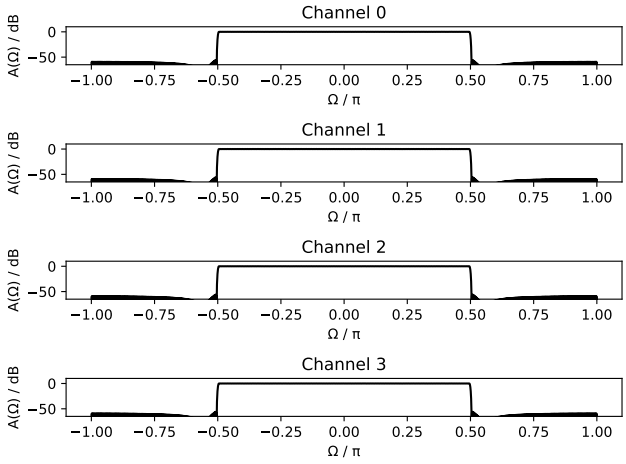


Figure 22: *The four (baseband) spectra of the outputs of the channelizer in Fig. 21.*

[5] J.C. Chuang W. Rhee and Jr L.J. Cimini. "Performance Comparison of OFDM and Multitone with Polyphase Filter Bank for Wireless Communications". In: *IEEE* (1998). DOI: 0-7803-4320-4/9.

[6] Fred Harris, Xiaofei Chen, and Elettra Venosa. *Cascade of Perfect Reconstruction Analysis and Synthesis Filter Banks: The New Architecture of Next Generation Wideband Receivers*. Tech. rep. San Diego, CA, USA: Electrical and Computer Engineering Department, San Diego State University.

[7] P. Stanner. "Wideband Receivers with Dynamic Subchannel Spacing using Polyphase FFT Filter Banks". University of Applied Sciences Munich, 2022.

[8] Fred Harris. *Selectable Bandwidth Filter Formed From Perfect Reconstruction Polyphase Filter Bank*. Tech. rep. San Diego, CA, USA: San Diego State University, 2010.

[9] Norbert Fliege. *Systemtheorie*. B.G. Teubner Stuttgart, 1991.

[10] R.E. Crochiere and L.R. Rabiner. *Multirate Digital Signal Processing*. Acoustics Research Department, Bell Laboratories, AT&T, Murray Hill, New Jersey.

[11] P. Murali Krishna and Babu T.P. Sameer. "Polyphase Channelizer Demystified". In: *IEEE Signal Processing Magazine* (2016).

[12] Deepa Kundur. "The Z-Transform and Its Properties". University of Toronto.

[13] Fred Harris. "Performance Options of a High Performance Receiver Filter Bank Channelizer". In: *IEEE* (2020). DOI: 10.1109/WPMC50192.2020.9309464.

[14] Deepa Kundur. "Discrete Time-Sinusoids". University of Toronto.

[15] Fred Harris, Xiaofei Chen, and Elettra Venosa. "An Efficient Based Spectrum Analyzer for Arbitrary Center Frequencies and Arbitrary Resolutions Analysis". In: *IEEE* (2011).

[16] Fred Harris et al. "Wideband 160-channel polyphase filter bank cable TV channeliser". In: *IET Signal Processing* (2010).

[17] Bernd Becker, Rolf Drechsler, and Paul Molitor. *Technische Informatik*. Pearson Studium, 2005.